System And Method For Reducing The Size Of RC Circuits

Field of Invention

[001] This invention relates in general to circuit analysis and in particular to a system and method for transforming a circuit from its original topology to a reduced topology.

Background of Invention

[002] Modern circuit simulations or verifications often encounter circuit models having a large number of inter-connected circuit elements. For example, during the final timing verification of Integrated circuits (IC), the number of parasitic resistors and capacitors in the circuit model may reach or exceed 10^7. In order to improve the efficiency of circuit analysis, a variety of RC reduction technologies have been developed during the last decade.

[003] The dominant trend in RC reduction has been towards transforming the original RC circuit network into a mathematical macro-model so that certain mathematical techniques can be utilized to generate a reduced mathematical circuit model. (See Rohrer et. al., "Asymptotic Waveform Evaluation for Timing Analysis", IEEE Trans. Computer Aided Design, vol. 9, pp. 352-66, 1990.) Such transformations also preserve the electrical properties of each input and output port. A disadvantage of these transformations, however, is that the reduced circuit model is not realizable. In other words, the reduced mathematical circuit model contains higher order elements that can not be realized as RC circuits. A drawback associated with this unrealizability is that the resulting mathematical models may presume special, often non-standard simulator capabilities.

[004] Liao et al. in "Partitioning and Reduction of RC Interconnect Networks Based on Scattering Parameter Macromodels", ICCAD 1995, pp. 704-09 (hereinafter "Liao") presented a realizable reduction scheme in which the original circuit model was partitioned into several sub-circuits. Each sub-circuit was then synthesized to produce a reduced sub-network that approximates the original sub-circuit. A weakness of the approach in Liao is that the reduction scheme may not be effective for circuits containing certain types of circuit topologies, such as coupling capacitors or an unbalanced tree in a memory block. Moreover, there is no guarantee that errors resulted from the approximations are controlled to a minimized or negligible level so that certain properties of the original circuit are preserved.

[005] Sheehan in "TICER: Realizable Reduction of Extracted RC Circuits", ICCAD 1999 (hereinafter "TICER") proposed a method of achieving realizable reduction of RC circuit with certain controlled accuracy in the neighborhood of an eliminated node. Details of Sheehan's method can be found in TICER, the content of which is incorporated hereby in its entirety as reference. Sheehan's method is based on node elimination. According to Sheehan, a node can be a quick, slow, or normal node depending on its time constant. The time constant of a node N is the total capacitance (χ_N) from the node to other nodes and to ground divided by the sum of conductance (γ_N) from the node to other nodes and to ground. Each node of a circuit is classified as a quick, slow, or normal node according to whether its time constant is less than, greater than, or between the minimum and maximum time constants defining the frequency range of interest. Sheehan concluded that quick nodes can be eliminated from the circuit network without significantly altering its behavior (such as delay characteristics or more specifically the Elmore delay) in the frequency range of interest.

[006] Sheehan then presented a time constant equilibration reduction (TICER) algorithm that reduces an RC circuit by successively finding quick nodes in the circuit and then eliminating them by the following procedures. To eliminate a quick node N, first remove all resistors and capacitors connecting other nodes to node N. Then insert new resistors and capacitors between former neighbors of node N according to the following two rules. If node i and node j had been connected to node N through conductance g_{iN} and g_{jN}, insert a conductance $g_{iN}g_{jN}/\gamma_N$ between nodes i and j. If node i had a capacitor c_{iN} connected to node N and node j had a conductance g_{jN} connected to node N, insert a capacitor $c_{iN}g_{jN}/\gamma_N$ between nodes i and j. A drawback of the TICER algorithm is that it only focuses on controlling the accuracy of circuit simulation and ignores the topological impact of node eliminations. This is because although TICER reduces the number of nodes, it may increase the number of resistors or capacitors between nodes neighboring the eliminated nodes. The increased number of resistors or capacitors could further complicate the entanglement between the remaining nodes. To further illustrate TICER's drawback, consider an example shown in Figure 1. Figure 1 shows a circuit fragment (or a "regional" circuit topology) comprising a quick node N having four neighboring nodes 1-4 connected to node N through four resistors with conductance $g_{1N} \ldots g_{4N}$. Eliminating node N according to TICER produces six new conductance connected between the four neighboring nodes. In general, eliminating a quick (or slow) node having M neighboring nodes could produce

M(M-1)/2 new resistors or capacitors between the neighboring nodes. These new resistors or capacitors could further aggravate the subsequent circuit analysis. As a result, TICER may not achieve significant circuit reduction for large circuit networks

[007] There is therefore a need to provide a method of producing a reduced topology that takes into account both the physical and topological characteristics of the circuit network. There is also a need to provide a system that implements the above method.

Summary of Invention

[008] The present invention includes a method of transforming a first topology to a reduced topology, said first topology representing an abstraction of one or more objects, said first topology further comprising a plurality of inter-connected elements. The method comprises the steps of: (a) identifying one or more elements for reduction; (b) analyzing the effect of reducing one or more of said identified elements on the topological and physical characteristics of said one or more objects, and (c) if the effect is negligible, generating a second topology reflecting the reduction of one or more identified elements.

[009] The present invention also includes a method of transforming a circuit from a first topology to a reduced topology, said first topology comprising a plurality of inter-connected circuit elements. The method comprises the steps of: (a) identifying one or more circuit elements; (b) analyzing the effect of reducing one or more of said identified circuit elements on the topological and physical characteristics of said circuit; and (c) if the effect satisfies a first standard, generating a second topology reflecting the reduction of one or more identified circuit elements. Steps (a)-(c) of the above methods may be recursively executed until no further reduction is possible.

[0010] The present invention further includes a system for transforming a first topology to a reduced topology, said first topology representing an abstraction of one or more objects, said first topology further comprising a plurality of inter-connected elements. The system comprises one or more software objects configured to (a) identify one or more elements; (b) analyze the effect of reducing one or more of said identified elements on the topological and physical characteristics of said one or more objects, or (c) generate a second topology reflecting the reduction of one or more identified elements, if the effect is negligible. Such system may further comprise one or more software objects configured to recursively executing (a), (b) and (c) until no further reduction is possible.

[0011] The present invention further includes a method of reducing coupling capacitors within a circuit topology. The method comprises the steps of: identifying one or

more C-blocks; producing a first regional topology approximating the effect of eliminating one or more of said C-blocks; estimating the coupling effects of one or more crossly-coupled nodes in the first regional topology; and generating a second regional topology replacing the cross-coupling of one or more crossly-coupled nodes with their corresponding estimated coupling effects.

[0012] The present invention further includes a method of transforming a circuit from a first topology to a reduced topology, said first topology comprising a plurality of inter-connected circuit elements. The method comprises the steps of: generating a tree-like topological approximation of at least one partition of the first topology; identifying one or more circuit elements for reduction from the tree-like topological approximation; and reducing one or more of said circuit elements in a bottom-up fashion from the leaf nodes to the root of the tree-like topological approximation.

Brief Description of Drawings

[0013] **Figure 1** illustrates an example circuit topology before and after node elimination according to the TICER algorithm.

[0014] **Figure 2** is a flow chart of one embodiment of the present invention.

[0015] **Figure 3** is a flow chart of another embodiment of the present invention.

[0016] **Figures 4A-4C** illustrates the TICER algorithm using an example circuit fragment having a coupling capacitor C_1 connected between two series of resistors R_1 R_2 and R_3 R_4.

[0017] **Figures 5A-5B** illustrates one way of estimating the coupling effect based on **Figure 4C**'s circuit.

[0018] **Figure 6** is a flow chart of yet another embodiment of the present invention.

[0019] **Figures 7A-7B** illustrates one embodiment of a method of reducing a cluster.

[0020] **Figure 8** is a flow chart of another embodiment of the present invention.

[0021] **Figure 9** is a flow chart of another embodiment of the present invention.

[0022] **Figure 10** is a flow chart of another embodiment of the present invention.

[0023] **Figure 11** is a flow chart of another embodiment of the present invention.

[0024] **Figures 12A-12B** illustrate a circuit transformation that results in a voltage divider structure.

[0025] **Figure 13** illustrates the step response of a neighboring node before and after eliminating a node whose degree of connectivity is equal to one.

[0026] **Figure 14** is a flow chart of another embodiment of the present invention.

[0027] **Figure 15** illustrates a regional topology identified for reduction.

[0028] **Figure 16** is a flow chart of yet another embodiment of the present invention.

[0029] **Figure 17** is a flow chart of yet another embodiment of the present invention.

[0030] **Figure 18** illustrates one embodiment of a method of reducing nodes.

[0031] **Figure 19** illustrates one embodiment of a method of reducing capacitors.

Detailed Description of Preferred Embodiments

[0032] One embodiment of the present invention comprises a method of transforming a first topology to a reduced topology. The first topology may represent an abstraction of one or more objects. The one or more objects can be circuits or any other objects. The first topology may further comprise a plurality of inter-connected elements. These inter-connected elements can be circuit elements, such as nodes, capacitors, transistors, or any other elements. The reduced topology may be obtained by eliminating elements or making approximations.

[0033] A flow chart of one method of the present invention is shown in Figure 2. Step 210 of the method identifies one or more elements for reduction. The one or more elements for reduction can be identified according to any applicable topological or physical standard. In one embodiment, symmetric nodes on the topology may be identified since these nodes may represent points or parts of the objects having similar behavior. Approximations of the first topology may be produced to facilitate the identification of elements for reduction. The approximation may be a minimum spanning tree (MST) of the first topology, or any other forms.

[0034] In step 220, the effect of reducing one or more of said identified elements is analyzed based on the topological and physical characteristics of said identified elements. The physical characteristics may include delay characteristics for circuit elements, or other characteristics for other objects. The topological characteristics may include connectivity of elements, topological complexity, or other characteristics.

[0035] In step 230, the method determines if the effect is negligible. The standard for determining the negligibility of the physical and topological effect may include any user-defined standard. For circuit networks, changes in physical effects, such as delay characteristics like Elmore delays are considered to be negligible if they are less than a certain percentage of the total delay. The topological effect may be determined by analyzing the complexity of topological approximations or equivalent topologies reflecting the reduction of one or more identified elements. For circuit networks, the topological effect may be considered to be negligible if it does not increase the complexity of the circuit topology. If the effect is determined to be negligible in step 230, the method proceeds to step 240.

[0036] In step 240, the method generates a second topology reflecting the reduction of one or more of the identified elements if it is determined that the effect is negligible. The second topology may be a reduced topology, or it may be a topology that facilitates further reduction. For example, a reduced topology may be generated by merging symmetric nodes. In one embodiment, steps 210-240 are executed recursively until no further reduction is possible. In another embodiment, the method may be executed again with more relaxed negligibility standards.

[0037] In one embodiment, step 210 comprises the step of producing a minimum spanning tree (MST) of the first topology, if the first topology is not a tree. The MST can be generated using any standard MST algorithm. In one embodiment, the MST is generated using the Prim's algorithm. A complete description of the Prim's algorithm and other MST algorithms can be found in M.A. Weiss, "Data Structures and Algorithm Analysis in C", 2nd edition, Addison-Wesley 1997, the contents of which are incorporated hereby in its entirety as reference.

[0038] For circuit networks, one or more input ports or drivers may be selected as the root nodes of the MST, or the first topology if it is a tree. In one embodiment, the MST or the first topology may be further divided into subtrees based on the root nodes. Some further approximations may be performed on the circuit topology before or after generating the MST. For example, several circuit elements may be merged into a macro-

node in order to simplify the analysis. In one embodiment, a macro-node is a maximal subcircuit with either a tree or a mesh structure. Furthermore, macro-nodes may be formed in such a way that they are connected in a tree manner. In another embodiment, the circuit network is first partitioned into one or more sub-nets. These one or more sub-nets may include one or more "DC nets". A DC net is a sub-net that is connected via capacitors to other neighboring sub-nets. More specifically, the following conditions may be used to identify a DC net. First, the DC net may include a node N, if there is at least one path of resistors from a root node to node N. The root node may be any node in the DC net. In one embodiment, a driver in the DC net is selected as a root node. Secondly, if a node N satisfies the first condition, then all ground capacitors connected to node N, as well as all resistors on any path from the root node to node N, are also included in the DC net.

[0039] In another embodiment of the present invention, small-valued circuit elements are reduced from the circuit topology. For example, a resistor may be considered to be small-valued if its resistance is less than a threshold, such as 0.001 Ω, 0.01 Ω, or 0.1 Ω. A capacitor may be considered to be small-valued if its capacitance is less than 0.1f F, 1fF, or 10fF. Eliminating small-valued circuit elements is relatively safe because most of the time it only negligibly affects the physical characteristics of the circuit. But, care must be taken in the elimination process to avoid further complicating the circuit topology.

[0040] A flow chart of one embodiment of the method of reducing small-valued circuit elements from the circuit topology is shown in Figure 3. Step 305 determines if the current circuit topology is a tree. If it is a tree, the method proceeds to step 320. If the current topology is not a tree, a MST of the current circuit topology is generated in step 310. In step 320, one or more small-valued circuit elements are identified from the tree or the MST. Eliminating these circuit elements may be safe because nodes connected to these circuit elements are likely to be quick nodes. Step 330 analyzes the variation of one or more delay measurements. This step further guarantees that the identified circuit elements can be eliminated without significantly altering the physical characteristics of the original circuit. Sheehan's formulae, for example, may be used in this step to estimate the change of Elmore delay after eliminating the identified circuit elements. Step 340 determines whether the variation is negligible. If it is negligible, step 360 eliminates those identified small-valued circuit elements. Step 350 determines whether there are any circuit elements remaining to be reduced. If so, control returns to step 320. In this manner, small-valued circuit elements can be recursively identified and eliminated from the tree or the MST until no more circuit element is available for reduction.

[0041] Yet another embodiment of the present invention comprises methods of reducing circuit elements, such as capacitors, that have coupling effects. Such coupling capacitors often exists between two series of resistors. Reducing these coupling capacitors can greatly simplify the circuit topology because it enables the series of resistors to be combined, thereby achieving further circuit reduction. Figure 4A illustrates an exemplary portion of a circuit fragment having a coupling capacitor C_1 connected between two series of resistors R_1 R_2 and R_3 R_4.

[0042] The prior art TICER algorithm cannot eliminate a coupling capacitor without further complicating the circuit topology. As an example, consider the circuit fragment shown in Figure 4A. Applying the TICER algorithm, eliminating C_1 involves eliminating nodes a and b. As shown in Figure 4B, according to TICER, two cross-coupling capacitors C_{1b} and C_{2b} are created after eliminating node a. Moreover, as shown in Figure 4C, two cross-coupling capacitors C_{23} and C_{14} are created following the elimination of node b. The creation of these two cross-coupling capacitors is undesirable since it further complicates the entanglement of circuit elements.

[0043] One embodiment of the present invention comprises a method of reducing the coupling effect of a circuit without creating any cross-coupling circuit elements. A flow chart of one embodiment of this method is shown in Figure 6. Step 610 identifies one or more circuit elements, such as coupling capacitors, having coupling effects. Step 620 produces a first regional topology approximating the effect of eliminating one or more circuit elements having a coupling effect. The first regional topology may be produced using any suitable mathematical or topological method. In one embodiment, coupling capacitors are reduced "C-block" wise. A C-block is a set of nodes connected via coupling capacitors. For example, nodes a, b in Figure 6 form a C-block. To eliminate the coupling capacitors connected between nodes in a C-block, nodes in the C-block are eliminated one after the other. The TICER algorithm may be used to approximate the effect of node elimination in a C-block . Step 630 identifies one or more crossly-coupled nodes (such as capacitors C_{14} and C_{23} in Figure 4C) within the first regional topology. Step 640 estimates the coupling effects of each said one or more crossly-coupled nodes. The coupling effect of these crossly-coupled nodes can be estimated based on the value of the capacitors in the first regional topology connected to the one or more crossly-coupled nodes. One way of estimating the coupling effect is described in the paragraph below. Step 650 generates a second regional topology by replacing the cross-coupling of each crossly-coupled node with its estimated coupling effects. Step 660 analyzes the effect of replacing the first regional topology with the second regional

topology. The effect can be determined, for example, based on the change of delay characteristics of other circuit elements. Step 670 determines whether the effect is negligible. If it is negligible, step 680 replaces the first regional topology with the second regional topology. Step 690 determines whether there are any circuit elements remaining to be reduced. If so, control returns to step 610.

[0044] Figure 5 illustrates one way of estimating the coupling effect. Figure 5's first regional topology contains four nodes 1-4 with coupling capacitors C_1, C_2 and cross-coupling capacitors C_3, C_4. The coupling effect of each node is estimated as follows. Let C_{min} be the minimum value between capacitance C_3 and C_4. For simplicity, assuming $C_{min} = C_3$, i.e. we assume $C_3 \leq C_4$ in Figure 5's circuit. Also let ΔC be the absolute value of the difference between C_3 and C_4. It is then estimated that the coupling effect of node 1 be a capacitance C_1+C_3 connected between nodes 1 and 3. The coupling effect of node 3 is estimated to be a capacitance C_1+C_3 connected between nodes 1 and 3 plus a capacitance ΔC connected between node 3 and the ground. Likewise, the coupling effect of node 4 is estimated to be a capacitance C_2+C_3 connected between nodes 2 and 4. The coupling effect of node N_2 is estimated to be a capacitance C_2+C_3 connected between nodes 2 and 4 plus a capacitance ΔC connected between node 2 and the ground. After estimating the coupling effects, the second regional topology is generated by replacing capacitors C_1-C_4 with the coupling effect of nodes 1-4 as estimated above.

[0045] In another embodiment of the present invention clusters of circuit elements are reduced from the circuit topology. A cluster typically comprises a series of resistors along with their end nodes and their surrounding circuit elements. Each of the surrounding circuit elements may connect between one node within the series of resistors and another node not within the series of resistors. Figure 7A shows an example cluster having a series of resistors R_5-R_7, four end nodes N_1, N_c, N_3, N_4 and surrounding elements including capacitors C_1-C_4 and resistors R_1-R_4. Clusters can be identified directly from the circuit topology or from the MST of the circuit topology. In one embodiment, clusters are identified in a bottom-up fashion from the MST. Clusters can be used to model memory blocks in a circuit and thus may appear in large numbers in circuit models representing micro-processor chips. Reducing clusters, therefore, can be beneficial at least to the subsequent analysis of micro-processor circuits.

[0046] A flow chart of one embodiment of the method of reducing clusters is shown in Figure 8. Step 810 identifies one or more clusters for reduction. Step 820

determines one or more center nodes for one or more clusters. A center node can be the topological center of the cluster, or it can be selected based on the value of resistors on its right or left hand side in the cluster, or, it can be any arbitrarily defined center node. Step 830 analyzes the effect of moving the connection of one or more circuit elements connected to one or more nodes within one or more said clusters to one or more said center nodes. The effect can be determined, for example, based on the change of delay characteristics of other circuit elements. Step 840 determines whether the effect is negligible. If it is negligible, step 850 moves the connection of one or more circuit elements within one or more said clusters to one or more said center nodes. Step 860 determines whether there are any circuit elements remaining to be reduced. If so, control returns to step 810.

[0047] The above method is further illustrated using Figure 7A's circuit. In Figure 7A's cluster, a center node N_c is first determined. After N_c is determined, the connection of capacitors C_1-C_4 and resistors R_1-R_7 are moved to N_c as shown in Figure 7B. Resistors R_6 and R_7 are also merged. Furthermore, resistor R_1-R_4 and capacitors C_1-C_4 can also be merged in many cases. For example, if resistors R_1-R_4 or capacitors C_1-C_4 are connected to a common node outside the cluster, these resistors or capacitors can be merged into one resistor or capacitor. In general, the effect of the above transformation on the Elmore delay is minimized when all resistors within the series of resistors are small-valued. However, when the cluster is connected between two transistors, the effect can still be negligible even if some or all resistors within the series of resistors are not small-valued.

[0048] Yet another embodiment of the present invention comprises methods for node reduction within a circuit topology. Figure 9 depicts a flow chart of this embodiment. In step 910, one or more nodes having similar input-output characteristics, such as similar input-output voltage or current, are first identified. In step 920, the effect of merging one or more of these nodes, such as the effect on the Elmore delay, is then analyzed. Step 930 determines if the effect is negligible. If so, step 940 merges one or more of these nodes into one node. Step 950 determines whether there are any circuit elements remaining to be reduced. If so, control returns to step 910.

[0049] In another embodiment of node reduction methods, one or more candidate nodes are first identified. These candidate nodes are likely to be quick nodes according to TICER and may therefore be safely eliminated without affecting the delay characteristics of the circuit network. However, care must be taken in applying TICER's node elimination rules to avoid further complicating the circuit topology. The following rules may be applied to ensure a reduced topology after eliminating one or more candidate nodes.

The general rule is that one or more candidate nodes can be eliminated only if eliminating them produces a less number of circuit elements connected between their neighboring nodes. Figure 10 depicts a flow chart of one embodiment applying the general rule. In step 1010, one or more candidate nodes are identified. Step 1020 produces a regional topology approximating the elimination of one or more candidate nodes. The TICER algorithm, for example, may be used to produce such an approximation. Step 1030 determines, based on the regional topology produced in step 1020, the difference in the number of circuit elements connected between the neighboring nodes of one or more candidate nodes, before and after eliminating one or more candidate nodes. Step 1040 determines if the regional topology produces less number of circuit elements. If so, step 1050 eliminates the one or more candidate nodes. Step 1060 determines whether there are any circuit elements remaining to be reduced. If so, control returns to step 1010.

[0050] Besides the general rule, several more specific rules may also be applied to facilitate the node elimination. First, a quick node N_c can be eliminated if its degree of connectivity is equal to two, i.e. there are only two resistors (and therefore only two neighboring nodes) connected to N_c. Applying TICER's rules, eliminating N_c only produces an equivalent resistor connected between the two nodes neighboring N_c. These quick nodes may be eliminated at the beginning of the reduction process so that small resistors connected to these nodes are first merged instead of being shorted. Figure 11 depicts a flow chart of one embodiment of the first rule. In step 1110, one or more candidate nodes are identified. These one or more candidate nodes may be quick nodes or other nodes of interest. Step 1120 determines the degree of connectivity of one or more candidate nodes. Step 1130 determines if the degree of connectivity is equal to two. If so, step 1140 eliminates the one or more candidate nodes. Step 1150 determines whether there are any circuit elements remaining to be reduced. If so, control returns to step 1110.

[0051] As a second rule, a candidate node whose degree of connectivity is equal to one may be eliminated (by, for example, shorting the resistor connected to the candidate node) only if the topology after the elimination does not contain a voltage divider structure comprising two capacitors connected in series. The reason behind this rule is that voltage divider structure may significantly alter the delay characteristics of the neighboring node. To further illustrate this point, consider the exemplary circuit fragment shown in Figure 12A. In Figure 12A's circuit, assuming R_3 is a small-valued resistor. Node N_2 therefore becomes a candidate node with a connectivity equal to one. The regional circuit topology after eliminating N_2 contains a voltage divider structure comprising capacitors C_1

and C_2, as shown in Figure 12B. (Note that eliminating N_1 can also resulting in a similar voltage divider structure.) Assuming a unit step input is applied between node N_a and the ground, the step responses at the neighboring node N_1 before and after eliminating N_2 are shown in Figure 13. It can be seen from Figure 13 that the voltage divider structure causes a voltage jump of $C_1/(C_1+C_2)$ volt at N_1. This voltage jump is often undesirable in the subsequent circuit analysis because it greatly alter the delay characteristics of node N_1. Also contained in the second rule is that nodes having capacitors directly connected to the input node of a circuit are kept from elimination. This is because eliminating such a node could generate a capacitor directly connected from the input to an output of the circuit, which will also cause an output voltage jump when an input voltage jumps.

[0052] Figure 14 depicts a flow chart of one embodiment of the second rule. In step 1410, one or more candidate nodes are identified. Nodes having capacitors directly connected to the input node of a circuit are excluded from candidate nodes. Step 1420 determines if the topology after eliminating one or more of said candidate nodes contains a voltage divider structure. Step 1430 determines if a voltage divider exists. If so, step 1440 eliminates the one or more candidate nodes. Step 1450 determines whether there are any circuit elements remaining to be reduced. If so, control returns to step 1410.

[0053] The third rule of node reduction comprises the step of identifying a regional topology comprising two nodes N_1 and N_2 connected to a common node N_a, wherein N_1 is connected to N_a through resistance R_1, N_2 is connected to N_a through resistance R_2, N_1 and N_2 further having capacitors C_1 and C_2, respectively, connected to either the ground or another common node. Moreover, there may not be other resistor connections to N_1 or N2. Furthermore, there may not be additional capacitor connections to N_1 or N2 either. An example of such a regional topology is depicted in Figure 15A. Experiments show that nodes N_1 and N_2 can be merged without significantly affecting the delay characteristics of the circuit network if the value of R_1*C_1 is approximately equal to the value of R_2*C_2. Furthermore, if N_1 and N_2 are further coupled to common nodes N_{b1}, N_{b2},...,N_{bn} via capacitors C_{c11}, C_{c12}, ...C_{c1n} and C_{c21}, C_{c22}, ...C_{c2n}, respectively, then nodes N_1 and N2 can still be merged if the value of R_1*Cc_{1i} is approximately equal to the value of R_2*Cc_{2i}, for i=1,...,n. An example of a regional topology where N_1 and N_2 are further coupled to a common node N_{b1} via capacitors C_{c11} and C_{c21} is shown in Figure 15B.

[0054] Figure 16 depicts a flow chart of one embodiment of the third rule. Step 1610 identifies a regional topology comprising two nodes N_1 and N_2 connected to a

common nodes N_a, wherein N_1 is connected to N_a through resistance R_1, N_2 is connected to N_a through resistance R_2, N_1 and N_2 further having capacitors C_1 and C_2, respectively, connected to either the ground or another common node. Step 1620 determines if the value of R_1*C_1 is approximately equal to the value of R_2*C_2 and if the value of R_1*Cc_{1i} is approximately equal to the value of R_2*Cc_{2i}, if N_1 and N_2 are connected to additional common nodes. If so, step 1630 merges nodes N_1 and N_2. Step 1640 determines whether there are any circuit elements remaining to be reduced. If so, control returns to step 1610.

[0055] The fourth rule of node reduction comprises the steps of identifying one or more symmetric nodes from a circuit topology comprising a single input tree structure and merging said one or more symmetric nodes if the effect of merging these nodes on the physical characteristics is negligible. The reason behind this rule is that symmetric nodes are likely to exhibit similar behavior. These nodes can therefore be merged without significantly affecting the delay characteristics of the original circuit network. The symmetric nodes can be identified and merged using any applicable algorithms. In one embodiment, the symmetric nodes are identified in a top-down fashion from the first tier to the last tier of the tree.

[0056] Figure 17 depicts a flow chart of one embodiment of the fourth rule. In step 1710, one or more symmetric nodes are identified from a circuit topology comprising a single input tree structure. This circuit topology may be a subcircuit of the original circuit topology. More specifically, in the original topology, a tree structure with multiple inputs ports may be divided into subtrees, each with a single input port. Step 1720 analyzes the effect of merging these nodes on the physical characteristics, such as Elmore delays, of other circuit elements. Step 1730 determines if the effect is negligible. If so, step 1740 eliminates the one or more symmetric nodes. Step 1750 determines whether there are any circuit elements remaining to be reduced. If so, control returns to step 1410.

[0057] The fifth rule of node reduction comprises the steps of identifying two nodes P_1 and P_2 satisfying the following conditions: (i) they are receiver ports with capacitor loads C_1 and C_2, respectively, (ii) they are connected to a common node A via resistances R_1 and R_2, respectively, and (iii) both $R_1 * C_1$ and $R_2 * C_2$ are small; and merging nodes P_1 and P_2. Figure 18 depicts one embodiment of the above method. In Figure 18A, nodes P_1 and P_2 are identified since they satisfy the above conditions (i)-(iii). In Figure 18B, nodes P_1 and P_2 are merged into one node P and their capacitors loads C_1 and C_2, as well as the two resistors R_1 and R_2 are combined.

[0058] The present invention further comprises methods of reducing capacitors connected between nodes. In one embodiment, a capacitor C connected between nodes A and B are eliminated if it is connected in parallel with a resistor R and the value of $\omega_{max} * C$ (ω_{max} equals 2π times the maximum operating frequency of the circuit) is far less then the value of R. In another embodiment, if a transistor drain or source node A is connected to a node B via a small resistor and node A and node B are coupled to node X via capacitors C_1 and C_2, respectively, then capacitor C_1 is moved from between nodes A and X to between nodes B and X and capacitors C_1, C_2 are then merged into one capacitor. Figure 19 depicts one embodiment of the above method. In Figure 19A, node A is the drain node of transistor 1910. If the resistor R connected between nodes A and B is small, then capacitor C_1 connected between nodes A and X is merged with capacitor C_2 connected between nodes B and X, as shown in Figure 19B.

[0059] Another embodiment of the present invention comprises a computer system including computer software program(s) that implement the above-described methods. The computer system may include a central processing unit (CPU) or other data processing means (e.g., plural processors), and a system memory for storing executable instructions and accessible data for the CPU or other processors. The system memory may take the form of DRAM (dynamic random access memory) and cache SRAM (static random access memory). Other forms of such memory may also be used. A system bus may be used to operatively interconnect the CPU and the system memory.

[0060] The computer system may further include non-volatile mass storage means such as a magnetic hard disk drive, a floppy drive, a CD-ROM drive, a re-writeable optical drive, or the like that is operatively coupled to the system bus for transferring instructions and/or data. Instructions for execution by the CPU (or other processors) may be introduced into the computer system by way of computer-readable media such as a floppy diskette or a CD-ROM optical platter or other devices.

[0061] The computer system may further include input/output (I/O) means to interface the system bus with peripheral devices such as display, keyboard and mouse. The I/O means may interface to a communications network such as an Ethernet network, a SCSI network, a telephone network, a cable system, or the like. Instructions for execution by the CPU and/or data structures for use by the CPU may be introduced into the computer system by way of data signals transferred over the communications network. The communication

network may therefore define a means for coupling to, and causing the computer system to perform operations.

[0062] The system memory may hold executing portions of an operating system (OS), such as Windows™ or Unix, and of any executing parts of one or more application programs. The application programs generally communicate with the operating system by way of an API (application program interface). The one or more application programs may include software program(s) adapted to execute one or more methods described in Figures 1-19 above. Such software programs may be written using any existing software programming language such as C or C++ and compiled to execute in the operating system.

[0063] In one embodiment, the software program(s) may comprise (a) software code to identify one or more elements for reduction; (b) software code to analyze the effect of reducing one or more of said identified elements based on the topological and physical characteristics of said identified elements, and (c) software code to generate a second topology reflecting the reduction of one or more of said identified elements, if the effect is negligible. The second topology can be a reduced topology, or a topology that facilitates further reduction. In another embodiment, the software program(s) may further comprise software code to recursively executing (a), (b) and (c) until no further reduction is possible.

[0064] The first topology may include graphical representation of one or more circuits. The plurality of inter-connected elements in the first topology may thus comprise a plurality of circuit elements. In one embodiment, the software program(s) further comprises software code to analyze the physical characteristics of the circuits. The physical characteristics may include the variation of one or more delay measurements, such as Elmore delays resulting from the transformation.

[0065] The above embodiments of the software program(s) can be implemented either separately or together in one or more software packages. Simulations have shown that for large circuit networks having more than 100 nodes, implementing one or more embodiments of the present invention can result in over 90% reduction of nodes and other circuit elements.

[0066] While the above invention has been described with reference to certain preferred embodiments, the scope of the present invention is not limited to these embodiments. One skilled in the art may find variations of these embodiments which,

nevertheless, fall within the spirit of the present invention, whose scope is defined by the claims set forth below.